

**IN THE CLAIMS**

Please cancel claim 20 without prejudice or disclaimer.

A complete claim listing is included beginning on the next page.

1. (original) A method operative in a content delivery network having a set of edge servers organized into regions, wherein the edge servers provide delivery of content on behalf of participating content providers, comprising:

for a given content provider, periodically generating a map identifying a set of alternate routes that may be used for edge server to content provider origin server communications;

in response to a request for a given file received at a given edge server, using the map to identify a direct route and a set of one or more alternate routes between the edge server and the content provider origin server;

executing a race by initiating a concurrent download of the file over each of the direct route and the set of one or more alternate routes; and

as a result of the race, determining an optimal route between the edge server and the content provider origin server.

2. (original) The method as described in claim 1 further including the step of using the optimal route for transferring additional data between the edge server and the content provider origin server for a given time period.

3. (original) The method as described in claim 1 wherein the map is generated by determining distances between a given set of regions and the content provider origin server.

4. (original) The method as described in claim 3 wherein the given set of regions are well-connected regions in the CDN.

5. (original) The method as described in claim 4 further including the step of collecting ping data between each well-connected region and the content provider origin server to facilitate generation of the map.

6. (original) The method as described in claim 1 further including the step of having the edge server fetch the map from a given location using a given name query.

7. (original) The method as described in claim 1 wherein the step of executing the

race further includes the steps of: determining whether a given number of bytes has been received; if the given number of bytes has been received over a given one of the routes, terminating download of the file over other of the given routes.

8. (original) The method as described in claim 1 wherein the given file comprises content that is not cacheable on the edge server.

9. (original) The method as described in claim 1 wherein the given file comprises content that is not then available on the edge server.

10. (original) The method as described in claim 1 wherein the given file comprises dynamic content.

11. (original) A method operative in a content delivery network having a set of edge servers organized into regions, wherein the edge servers provide delivery of content on behalf of participating content providers, and wherein a given content provider origin server may be reached over a set of routes, comprising:

identifying one or more routes between the edge server and the content provider origin server;

in response to a request for a given file received at a given edge server, initiating a concurrent download of the given file over each of the routes; and

determining an optimal route between the edge server and the content provider origin server as a function of the concurrent downloads.

12. (original) A server for use in a content delivery network, comprising:  
code executable in the server for initiating a performance metric test on a set of potential routes between the server and a given second server, wherein at least one of the potential routes passes through a server intermediate the server and the given server; and  
code executable in the server for collecting and analyzing data generated as a result of the performance metric test and determining an optimal path between the server and the given server.

13. (original) The server as described in claim 12 wherein the server is an edge server and the given server is a content provider origin server.

14. (original) The server as described in claim 12 wherein the server is an edge server and the given server is a reverse proxy server.

15. (original) The server as described in claim 12 wherein the performance metric test is a download of a file, wherein the file is an object being requested by an end user or a test object.

16. (original) The server as described in claim 15 further including code executable in the server for determining when a given number of bytes of the file have been received over a given one of the potential routes and for terminating the performance metric test with respect to other of the potential routes when the given number of bytes of the file have been received.

17. (original) The server as described in claim 12 further including code for initiating the performance metric test is responsive to a given client request.

18. (original) The server as described in claim 12 further including code for initiating the performance metric test if no such test has occurred in a given first time period, if a given second time period has elapsed since a last test, or if a given number of requests for the file have been received at the edge server without a test being run.

19. (original) The server as described in claim 12 further including code executable in the server for fetching a map defining the set of potential routes, wherein the map is generated on a per-content provider basis.

20. (canceled)